

画像生成 AI によるアニメキャラクター画像の 実写化と顔角度推定

枘井 暢久^{†1} 石川 颯志^{†1} 山西 良典^{†2} 仲田 晋^{†1}

概要: 本研究では、アニメキャラクターの顔画像から顔角度を推定することを目的とする。そのための手法として今回は、アニメキャラクターの顔画像を元に StableDiffusion と ControlNet を用いて生成した実写風顔画像を、実写顔画像の顔角度推定モデルで角度推定することによってアニメキャラクター画像の顔角度を推定する手法を提案する。アニメキャラクター画像の顔角度が推定できれば、アニメ映像の分析に貢献し、将来的なアニメ制作の支援につながる事が期待できる。

1. はじめに

日本のアニメーションは、日本国内だけでなく世界中の人からも人気を集めるコンテンツとなっており、日本を代表する文化として発展してきた。このアニメへの関心はアニメ作品そのものだけでなく、アニメの制作過程にも目が向けられている。2019 年に大英博物館で実施された「大英漫画展」[1] においても、漫画からアニメへの変換過程が紹介されている。我々はアニメを視聴する際に、日本の作画アニメと PIXAR などが制作するコンピュータグラフィックス (CG) アニメに対して描画や動きの差異を感性として認識できる。しかし、それらの差異について客観的指標に基づいた比較や検証として確立された論考は著者らの知る限り存在しない。作画アニメと CG アニメの差異を特徴づける観点として、色や影のつけ方 (例えば文献 [2][3]) や物理特性に沿わない動きのモデル化 [4] が注目されてきた。本稿では、この他の観点として構図に目を向けて、特にキャラクターの顔の描かれ方に着目する。作画アニメではキャラクターのセリフや表情によって物語が駆動していくため、キャラクターの顔はアニメ作品の中でも多くのシーンで描画されている。どのような方向を見たキャラクターがどのような画角で描かれているのかについて客観的に分析するためには、キャラクターの顔角度を推定する必要がある。しかしながら、顔画像から顔の角度や向きを取得するという研究において、アニメキャラクター画像を対象とした研究は未だ少ない。そこで、キャラクターの顔の描かれ方の分析のための要素研究として、アニメキャラクターの顔角度の推定に取り組む。

アニメキャラクターを対象とした顔角度推定の取り組みが少ない一方で、一般的な実写の人間に対する顔角度推定の研究にはこれまで多くの人に取り組んでおり、様々な推定手法が提案されている [5]。Hempel らが提案した 6DRepNet モデル [6] は、Ground Truth データに回転行列を適用し、6 次元回転行列で作成されたネットワークによってそれまでの手法よりも精度の高い推定を実現している。

また Ranjan ら [7] は、顔の角度だけでなく、顔の検出やランドマークの特定、性別認識も同時に行う HyperFace というマルチタスク深層学習手法を提案している。Malakshan ら [8] は、顔の超解像技術 (FSR) を利用して低解像度の顔画像に対する顔角度推定の精度を大幅に向上させた。このように実写の顔画像に対する顔角度推定の研究は数多く存在するが、アニメキャラクターの顔自体が特殊な描かれ方 (例えば、目が非常に大きい、横顔で口が真横の頬部に描かれるなど) で表現されるという特性を要因として、アニメキャラクターの顔角度推定に実写の人間に対する顔角度推定をそのまま用いることはできない。

本研究の目的は、アニメキャラクターの顔画像から顔の角度を推定することである。具体的には、アニメキャラクターの顔画像を入力とし、そのキャラクターの顔の向きを表すヨー・ピッチ・ロールの 3 つの回転角を出力することを目的としている。

この目的を実現するための方法として、アニメキャラクター画像と角度情報のペアデータを学習することで顔角度推定モデルを作成する手法が考えられる。しかし、アニメキャラクター画像に対して角度情報を取得する確実な手法はなく、そもそも描き手自身も正確な顔角度を意識して描いているとは限らない。つまり角度情報付きのアニメキャラクター画像を収集することは難しく、このアプローチは現実的ではないと考える。

そこで本研究では、アニメキャラクター画像を、画像生成 AI を用いて顔の向きを維持したまま実写風の画像に変換し、その画像に対して実写顔画像用の顔角度推定技術で角度推定する方法を提案する。前述のように実写顔画像に対しては精度の良い角度推定モデルが存在するため、アニメキャラクター画像を実写風画像に変換することで角度推定の精度向上を期待するという考え方である。この場合、顔の姿勢を維持したまま実写風画像に変換することが要求される。私たちは画像生成 AI の Stable Diffusion、および出力画像の構図や姿勢を制御できる ControlNet を活用するこ

^{†1} 立命館大学

^{†2} 関西大学

とでアニメキャラクター画像の姿勢を維持したまま実写風の顔画像を生成できると考えた。本研究の目的が達成できれば、アニメ映像の分析に貢献し、アニメ映像に対する理解が深まることで将来的なアニメーション制作の支援につながることを期待される。

2. 実写風画像の生成方法

本研究の目的は、アニメキャラクター画像の顔角度を推定することである。前述のように、私たちは顔角度を推定したいアニメキャラクター画像を元に、同じ向きを向いた実写風の人物の顔画像を作ることができれば、その画像を既存の実写人物用の顔角度推定モデルを用いて角度推定することで元のアニメ画像の顔角度を推定できるのではないかと考えた。

ここで課題となるのは、どのようにしてアニメキャラクター画像から顔の向きが同じ実写風の人物画像を作成するかである。この課題に対して私たちは、Diffusion Model (拡散モデル) に基づいた画像生成 AI の一つである Stable Diffusion [9] と、生成画像の構図や人物の姿勢を制御する機構である ControlNet [10] を組み合わせた画像生成によって解決できると考えた。ここでの Stable Diffusion に期待する役割は実写風画像への変換であり、モデルとプロンプトの選択により適切な出力を目指す。一方の ControlNet に期待する役割はキャラクター顔画像の姿勢の維持であり、入力キャラクター画像から ControlNet の構図情報を取得することで、そのキャラクターと同じ方向を向いた実写風の人物画像の生成を試みる。以降、具体的な画像の作成手順について述べていく。

Stable Diffusion による画像生成では Dreambooth [11] や LoRA [12] のような追加学習モデルを用途に応じて使い分けることが一般的である。本研究では写実的な人物画像の生成に特化したモデルを利用することで目的にあった画像変換が実現できると考える。

アニメキャラクターの姿勢維持のために利用する ControlNet に与える構図情報には複数の選択肢がある。バージョン 1.1 では入力画像から認識した人体の姿勢の情報を元に生成する OpenPose, 入力画像から抽出した輪郭線を元に生成する canny, 入力画像から抽出した深度情報を元に生成する depth など 14 種類の追加情報が選択肢として用意されており、このうち一つまたは複数を使用する設計となっている。

本研究ではキャラクター頭部の姿勢維持が必要要件であるため、キャラクター画像の輪郭を構図情報として与えられる canny, キャラクター画像から推定した深度値を構図情報として与えられる depth が有力候補と考えられる。また、入力画像をより高解像度な画像に変換する tile も写実性を高める効果が期待され、canny や depth と組み合わせ

ることで顔角度推定の精度向上に貢献することが期待できる。一方、人体の姿勢を反映できる OpenPose については頭部の姿勢維持に役立つ可能性はあるが、そもそもアニメキャラクター画像における OpenPose の精度が期待できないため本研究の目的には適さないと考えられる。以上のように、ControlNet の機能の選択と組み合わせは角度推定に影響すると考えられるため、いくつかの組み合わせについて次章にて推定精度を検証する。

3. 検証

アニメキャラクター画像の顔角度推定がどの程度正しく機能するかを検証する。なお、キャラクター画像を Stable Diffusion により実写風画像に変換するために、本稿ではフォトリアル系に特化した BracingEvoMix [13] と呼ばれるモデルを利用している。

3.1 検証用画像

顔角度の推定精度を検証するためには正確な顔角度情報が付与されたアニメキャラクター画像を用意する必要がある。また、アニメキャラクター画像は作品や個々のキャラクターの絵柄が角度推定の精度に影響すると考えられるため、検証に利用する画像も幅広いバリエーションを用意することが理想的である。その一方で、既存のアニメキャラクター画像の角度情報を得ることは難しく、実際のアニメ映像を検証に利用することができないという問題もある。そこで、アニメキャラクターを模した 3D モデルを用意し、指定した顔角度に応じた姿勢でレンダリングすることで角度情報付きの検証用画像として利用することとする。具体的には、3D キャラクター制作ソフトウェアである VRoid Studio [14] を用いて髪型、輪郭、眉・目・口・鼻の形にバリエーションを持たせた 10 体のキャラクターモデルを作成し、そのモデルを 3DCG の統合開発環境である Blender 上でシーンを設定後にセル画風のレンダリングすることで角度情報付きの画像を得ることとした。カメラとキャラクターの位置関係は指定した顔角度に応じて Blender スクリプトで自動設定している。また、ここでのセル画風レンダリングは輪郭線の描画とスムーズシェーディングの無効化を意味する。Blender での輪郭線描画は複数の方法があるが、ここでは Blender の標準機能である Freestyle を利用している。また、シェーディングの無効化は光源を環境光のみとすることで、面の向きによらずマテリアルの色をそのまま反映させることを意図している。これにより CG 特有のグラデーションを抑制しつつ、セルアニメの特徴である線画と塗りで構成される絵柄に近づける効果を期待する。

この手順で作成した 10 体のキャラクターモデルのレンダリング結果を図 1 に示す。この例ではキャラクターデザインのバリエーションを一定表現できており、かつ、線画



図1 Vroid Studio [14] で作成した 10 体の 3D モデルのレンダリング画像例



図2 検証で利用する 65 種類の顔角度 (左端がヨー回転 90°, 右端がヨー回転 -90°, 上端がピッチ回転 30°, 下端がピッチ回転 -30° を表す.)



図3 ヨー方向 90°, 75°, 60°, 45°, 30°, 15°, 0°, ピッチ方向 0° における「canny + tile」での実写風生成画像

と塗りで構成されるという意味でセル画の画風をある程度表現できているといえる。一方で、極端なデフォルメを伴うキャラクターを表現できない、表情が固定されている、絵柄が画一的という点において実際のアニメ映像における角度推定の評価として不十分な面もあるが、実写風への変換による角度推定精度の向上についての知見を得ることは期待できるため、本稿においてはこれを検証用のデータとして採用する。

検証用画像の角度は、ロールは 0° で固定、ヨーが -90°, -75°, -60°, -45°, -30°, -15°, 0°, 15°, 30°, 45°, 60°, 75°, 90° の 13 パターン、ピッチが -30°, -15°, 0°, 15°, 30° の 5 パターンの計 65 種類の角度 (図 2) で推定精度を検証することとする。これを 10 体に対して行い、合計 650 枚の検証用画像を用意した。

3.2 CG での角度推定精度の検証

650 枚の検証用画像に対してそれぞれ Stable Diffusion と ControlNet による実写風画像へ変換したのち、前述の顔角度推定技術による角度推定を行い、その精度を検証する。実写風画像を生成する際の ControlNet に与える構図情報としては前述のように depth と canny が有力な候補であり、また tile による画質向上の併用など複数の組み合わせな

表 1 レンダリング画像と生成した実写風画像の角度ごとにおける角度推定誤差の平均 (度)

「(A) レンダリング画像」での角度推定の平均誤差

		ヨー(度)												
		90	75	60	45	30	15	0	-15	-30	-45	-60	-75	-90
ピッチ(度)	30	75.6	67.3	71.2	62.6	58.4	61.1	64.9	65.1	80.4	92.0	104.4	115.3	128.6
	15	51.1	43.1	54.9	45.5	49.9	51.8	53.3	61.5	71.1	85.4	97.4	110.7	133.8
	0	46.1	40.8	44.0	41.9	41.0	40.2	42.1	49.3	59.1	78.1	89.6	104.8	130.0
	-15	60.3	44.6	38.6	33.0	29.2	29.6	30.7	38.0	49.2	64.7	83.6	104.1	122.9
	-30	54.8	41.0	31.1	19.4	15.3	17.0	24.0	33.0	44.2	65.2	79.4	99.9	119.5

「(B) depth 単体での実写風生成画像」での角度推定の平均誤差

		ヨー(度)												
		90	75	60	45	30	15	0	-15	-30	-45	-60	-75	-90
ピッチ(度)	30	41.2	38.1	35.9	33.7	30.2	25.3	22.2	25.4	22.8	27.4	33.9	40.7	74.4
	15	34.4	30.0	24.9	20.1	16.7	14.6	13.6	16.5	19.9	20.3	30.8	34.4	60.3
	0	24.7	19.5	21.3	18.3	12.8	8.9	12.0	11.6	12.6	16.6	24.0	31.7	43.4
	-15	25.4	15.5	21.9	20.4	13.9	12.3	9.6	7.6	12.5	17.9	34.6	18.7	36.1
	-30	31.2	24.9	26.2	33.5	29.0	17.7	17.3	12.5	17.2	26.6	29.2	65.4	74.3

「(C) canny + tile での実写風生成画像」での角度推定の平均誤差

		ヨー(度)												
		90	75	60	45	30	15	0	-15	-30	-45	-60	-75	-90
ピッチ(度)	30	61.7	61.7	32.2	23.7	21.9	21.3	23.3	25.3	25.3	28.3	34.5	57.1	74.5
	15	40.3	35.3	28.6	22.3	18.6	19.4	22.2	22.4	17.0	25.0	31.9	38.3	54.0
	0	33.8	27.4	22.0	20.6	17.4	15.8	15.5	16.3	16.9	19.5	23.2	25.5	38.0
	-15	34.2	23.6	17.0	14.4	10.7	9.6	6.3	8.3	10.5	14.7	16.8	22.2	40.9
	-30	39.4	22.8	11.5	9.1	7.1	4.7	7.3	5.1	7.9	9.2	16.2	26.6	37.5

表 2 「canny+tile」による生成画像の推定結果における回転角ごとの推定誤差の平均 (度)

「(C) canny + tile での実写風生成画像」でのヨー方向の平均誤差

		ヨー(度)												
		90	75	60	45	30	15	0	-15	-30	-45	-60	-75	-90
ピッチ(度)	30	45.1	47.9	15.6	10.8	5.9	1.4	1.6	2.5	3.4	8.8	15.9	42.5	61.6
	15	30.0	19.9	16.1	11.9	8.8	2.9	1.1	3.0	4.6	11.0	17.6	22.3	44.4
	0	28.0	17.3	14.3	12.3	7.0	2.4	1.2	2.4	4.8	11.5	14.5	16.9	32.5
	-15	35.0	23.1	14.6	10.4	4.7	1.3	1.0	2.2	6.2	12.0	13.3	19.7	41.7
	-30	42.3	25.7	13.1	10.0	7.1	2.0	3.7	4.0	6.8	9.4	17.5	30.0	41.4

「(C) canny + tile での実写風生成画像」でのピッチ方向の平均誤差

		ヨー(度)												
		90	75	60	45	30	15	0	-15	-30	-45	-60	-75	-90
ピッチ(度)	30	41.9	40.6	28.4	21.1	21.1	21.2	23.2	25.1	25.1	26.9	30.6	38.4	42.2
	15	26.1	28.3	23.6	18.5	16.1	19.0	22.1	22.2	16.1	22.3	26.4	30.9	29.7
	0	19.1	21.4	16.5	16.4	15.8	15.6	15.4	15.9	16.1	15.6	18.0	18.9	20.0
	-15	6.7	7.8	9.5	10.3	9.4	9.5	6.1	7.4	8.3	8.8	10.4	12.1	10.1
	-30	8.8	3.1	2.3	2.4	2.9	4.0	5.7	2.6	3.5	3.6	5.3	7.0	9.2

「(C) canny + tile での実写風生成画像」でのロール方向の平均誤差

		ヨー(度)												
		90	75	60	45	30	15	0	-15	-30	-45	-60	-75	-90
ピッチ(度)	30	13.3	8.6	6.4	5.8	4.3	2.7	0.8	3.7	6.0	10.9	11.6	9.7	10.3
	15	9.3	12.3	9.9	5.8	3.2	3.1	0.5	3.4	4.6	6.9	9.7	10.7	6.3
	0	13.6	14.9	10.8	5.0	4.6	3.5	0.8	4.5	6.0	5.1	6.8	8.8	9.1
	-15	12.2	9.8	8.0	5.5	4.6	4.0	0.7	4.5	3.8	2.4	4.3	9.3	9.1
	-30	8.1	9.7	9.4	5.7	3.9	4.2	2.1	3.0	4.0	4.7	6.0	6.3	6.0

ど複数の選択肢がある。これらの選択肢のうち本稿では「depth 単体」と「canny + tile」の 2 パターンによる実写風生成画像での結果を例示する。図 3 は「canny + tile」による変換の一部である。後述するように実写風変換は必ずしも良い結果が得られるわけではないが、頭部の姿勢はおおむね維持されていると考えられる。

角度推定には 6DRepNet を用いた。6DRepNet は、実写顔画像データセットである AFLW2000 と BIWI に対して、それぞれヨーの平均角度誤差が 3.63°, 3.24°, ピッチの平均角

度誤差が 4.91°, 4.48° と高精度での推定が可能との報告があり [6], 本研究における推定にも有用と考えた。

以下, 「(A) レンダリング画像」, 「(B) depth 単体での実写風生成画像」, 「(C) canny + tile での実写風生成画像」の 3 種類で角度推定することとする。10 体で角度推定したときの顔の向きを平均誤差を表 1 に示す。ここでの顔の向きの誤差とは, 顔の正面方向の 3 次元ベクトル \mathbf{v}_1 と, 推定された顔の向きの 3 次元ベクトル \mathbf{v}_2 のなす角 θ , つまり $\theta = \arccos(\mathbf{v}_1 \cdot \mathbf{v}_2 / (\|\mathbf{v}_1\| \|\mathbf{v}_2\|))$ のことである。

表 1 の「(A) レンダリング画像」, つまり実写風に変換せずに角度推定した結果に着目すると, 例えばヨーが 0°, ピッチが 0° での推定角度の誤差は 42.1° である。これは CG の正面画像をそのまま角度推定したときに 10 体平均で 42.1° もずれるという意味である。さらに顔の向きが上方向や横方向に傾くと誤差はさらに増加し, 顔の向きによっては推定角度の誤差が 90° を超える。これは顔角度推定が CG の画像では機能しないことを意味しており, アニメのキャラクター画像においても角度推定の精度は期待できない。

提案手法における角度推定の誤差, つまり表 1 「(B) depth 単体での実写風生成画像」と表 1 「(C) canny + tile での実写風生成画像」に着目する。この 2 つの推定角度の平均誤差は, 表 1 上段の「(A) レンダリング画像」の場合に比べて大きく減少していることがわかる。これは実写風生成画像の利用による精度改善の効果が大きいことを意味しており, 本稿の提案手法が効果的に機能しているといえる。特にヨー方向が 0° 前後かつピッチ方向が -15° 前後では精度が高く, 誤差が 10° 未満に抑えられることもある。「(B) depth 単体での実写風生成画像」と「(C) canny + tile での実写風生成画像」では誤差の数値は異なるものの, その傾向には大きな差が見られず, ControlNet での構図情報の選択の影響は少ないという結果となった。

一方, 顔の向きが上方向や横方向に傾くと誤差が増大する問題については「(B) depth 単体での実写風生成画像」や「(C) canny + tile での実写風生成画像」でも同様であり, 残念ながら提案手法の顔角度推定は実用的とは言えない。特に本稿で採用した角度推定手法 6DRepNet の誤差が 5° 未満 [6] であることを考えると実写風画像の生成過程で誤差の要因があることが考えられる。以下, この要因を探るための分析を進める。

角度推定誤差の傾向を理解するために, ヨー方向・ピッチ方向・ロール方向のそれぞれの推定誤差を分析することとする。表 2 は「(C) canny + tile での実写風生成画像」での実写風変換での角度推定誤差を表している。なお, 「(B) depth 単体での実写風生成画像」の推定誤差については大きな傾向の違いはなかったため本稿では省略している。この表より, ヨー方向・ピッチ方向・ロール方向のいずれも顔の向きと誤差の大きさには一定の傾向がみられる。表 2 上

段のヨーの推定誤差は特に顔の向きが横方向に傾くほど大きくなる傾向がある。顔が横を向くと実写風生成画像がうまくいかないという可能性が考えられるが, 図 3 を見る限りそうした傾向はみられず, 原因を明らかにするためにはさらなる分析が必要である。表 2 中段のピッチの推定誤差は特に顔が上方向に傾くほど推定誤差が大きくなる傾向がある。これは, 実写風画像に変換する際に上向きの画像の生成がうまくいかなかったためと考えられる。実際にレンダリング画像と生成した実写風画像を見比べると (図 4), 顔が上向きの画像はレンダリング画像と実写風画像でずれが大きいことが分かる。レンダリング画像のあごと口の線画消えているため, 実写風画像の生成に影響した可能性がある。表 2 下段のロール方向の推定誤差にも顔の向きとの関係が見てとれるが, ヨーとピッチの誤差に比べると影響は小さい。

3.3 実際のアニメ画像での角度推定

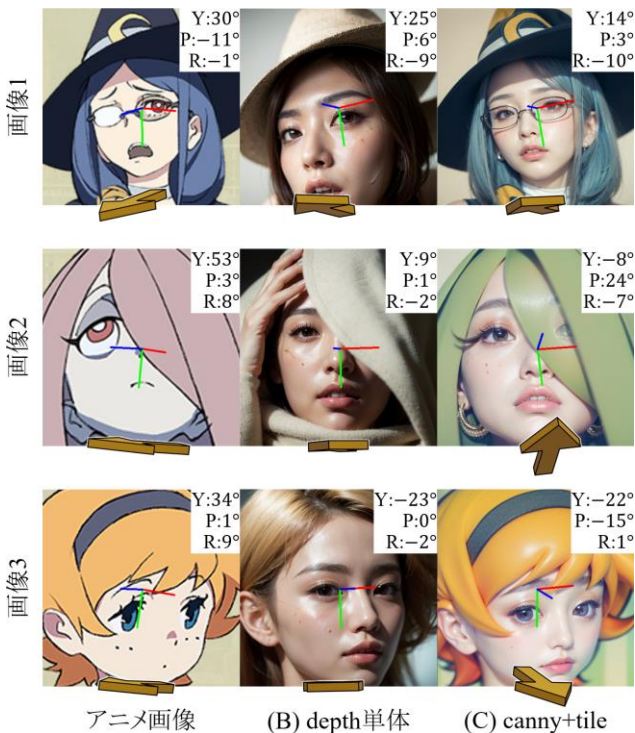
次に, 実際のアニメ画像を入力として角度推定を行った。入力には, 映画「リトルウィッチアカデミア 魔法仕掛けのパレード」[15] のキャラクター画像を使用した。それらの画像を「アニメ画像」, 「(B) depth 単体での実写風生成画像」, 「(C) canny + tile での実写風生成画像」の 3 パターンでそれぞれ 6DRepNet での顔角度推定を行った。その結果を図 5 に示す。図中の Y,P,R はそれぞれヨー, ピッチ, ロール方向の角度の推定値であり, 画像下部の矢印は推定された方向を表している。この例では正しい角度がわからないため誤差を求めることができないが, 推定角度の妥当性はある程度主観的に判断できる。「アニメ画像」で推定された向きはキャラクターの向きとは大きく異なっており, 角度推定が機能していないといえる。「(B) depth 単体」についてもこの例では本来のキャラクターの向きと大きく異なる方向が検出される結果となった。一つの要因としては実写風画像の生成段階で頭部の姿勢を維持できていないことが挙げられる。「(C) canny + tile」では画像 1 と画像 3 に関しては良い推定ができておりといえる。画像 2 に関しては頭部の左右の傾きがやや不正確に検出されているが, ピッチの推定値 (P:24°), つまり頭部の上下の傾きは適切に検出できていると思われる。全体として誤差の要因については現時点では不明な点が多く, さらなる分析と改善を進めたい。

4. まとめ

本研究の目的はアニメキャラクターの顔画像から顔の角度を推定することであった。この目的を達成するために, 本研究ではアニメキャラクター画像を Stable Diffusion と ControlNet を用いて顔の向きを維持したまま実写風の画像に変換し, その画像に対して実写顔画像用の顔角度推定技術で角度推定する方法を提案した。アニメキャラクターを



図 4 ピッチ方向 $-30^{\circ} \sim 30^{\circ}$ の実写風変換例



映画『リトルウィッチアカデミア 魔法仕掛けのパレード』
©2015 TRIGGER/吉成曜/GOOD SMILE COMPANY 協力:株式会社トリガー

図 5 実際のアニメ画像 [15] での角度推定結果

模した CG のレンダリング画像を用いて角度推定の検証を行った結果、アニメ画像をそのまま角度推定するよりも、実写風画像に変換して推定した場合の方が推定誤差が小さくなることが明らかになった。しかしながら、顔の向きによって推定結果に大きな差があり、実用的な推定精度とは言えない結果となった。今後の展望として、推定誤差の原因をさらに分析し、推定精度の改善を進めたい。

謝辞

論文執筆では、株式会社 TRIGGER 取締役舛本和也氏が

らの助言と協力を得た。記して謝意を表す。なお、本研究は JSPS 科研費 22K12338 の助成を受けたものである。

参考文献

- 1) “大英博物館でのマンガ展「The Citi exhibition Manga」レポート”. <https://mediag.bunka.go.jp/article/article-15688/> (参照 2023-7-19)
- 2) Chen, J., Liu, G. and Chen X.: AnimeGAN: A Novel Lightweight GAN for Photo Animation, Artificial Intelligence Algorithms and Applications, ISICA, pp. 242–256 (2019).
- 3) 山下 耀, 高野 裕基, 斎藤 隆文: “法線制御によるアニメーション風 CG のための陰影付け手法”, 情報処理学会第 77 回全国大会, pp63–64 (2015).
- 4) Furukawa, K. and Nakata, S.: Automatic Generation of the Periodic Hair Motion of 3D Characters for Anime Production, Entertainment Computing ICEC 2018, Lecture Notes in Computer Science 11112, pp. 289–292 (2018).
- 5) Asperti, A. and Filippini, D.: Deep Learning for Head Pose Estimation: A Survey, SN Computer Science, Vol.4, No.3, 349 (2023).
- 6) Hempel, T., Abdelrahman, AA. and Al-Hamadi, A.: 6d Rotation Representation for Unconstrained Head Pose Estimation, 2022 IEEE International Conference on Image Processing (ICIP), IEEE, pp. 2496–2500 (2022).
- 7) Ranjan, R., Patel, VM. and Chellappa, R.: Hyperface: A Deep Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(1), pp. 121–135 (2017).
- 8) Malakshan S. R., Saadabadi, M. S. E., Mostofa, M., Soleymani, S. and Nasrabadi, N. M.: Joint Super-Resolution and Head Pose Estimation for Extreme Low-Resolution Faces, IEEE Access, Vol.11, pp. 11238–11253 (2023).
- 9) Rombach, R., Blattmann, A., Lorenz, D., Esser, P. and Ommer, B.: High-Resolution Image Synthesis with Latent Diffusion Models, IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10684–10695 (2022).
- 10) Zhang, L. and Agrawala, M.: Adding Conditional Control to Text-to-Image Diffusion Models, arXiv preprint arXiv:2302.05543 (2023).
- 11) Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M. and Aberman, K.: Dreambooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation, IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 22500–22510 (2023).
- 12) Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. and Chen, W.: Lora: Low-Rank Adaptation of Large Language models, arXiv preprint arXiv:2106.09685 (2021).
- 13) sazyou_roukaku: sazyou-roukaku/BracingEvoMix, <https://huggingface.co/sazyou-roukaku/BracingEvoMix> (参照 2023-7-24)
- 14) pixiv, VRoid Studio, <https://vroid.com/studio> (参照 2023-7-24)
- 15) 映画『リトルウィッチアカデミア 魔法仕掛けのパレード』©2015 TRIGGER/吉成曜/GOOD SMILE COMPANY, 2015